(4)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AIM 1173	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A comparison of Hardware Implementations for Low-Level Vision Algorithms		5. TYPE OF REPORT & PERIOD COVERED memorandum 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ed Gamble		8. CONTRACT OR GRANT NUMBER(s) DACA76-85-C-0010 N00014-85-K-0124
9. PERFORMING ORGANIZATION NAME AND ADDRESS Artificial Intelligence Laboratory 545 Technology Square Cambridge, MA 02139		10. PROGRAM ELEMENT PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209		12. REPORT DATE November 1989 13. NUMBER OF PAGES 49
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Information Systems Arlington, VA 22217		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Analog VLSI, DSP
Subthreshold CMOS
Vision Hardware
Low-level vision
Real-time Image Processing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Abstract: Early and intermediate vision algorithms, such as smoothing and discontinuity detection, are often implemented on general-purpose serial, and, more recently, parallel computers. The excessive time required by these general-purpose computers prevents real-time computation of these vision algorithms. Special-purpose hardware implementations of low-level vision algorithms may be needed to achieve real-time processing. (continued on back)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20 continued:

This memo reviews and analyzes some hardware implementations of low-level vision algorithms. Two types of hardware implementations are considered: the digital signal processing chips of Ruetz (and Broderson) and the analog VLSI circuits of Carver Mead. Both these approaches claim to achieve real-time image processing; both have limited the vision problem that they solved in ways largely inconsistent with vision processing in unrestricted environments. The advantages and disadvantages of these two approaches for producing a general, real-time vision system are considered. As early attempts at comprehensive vision hardware, these two approaches provide useful insights for future developments of vision hardware.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1173 November 1989

A COMPARISON OF HARDWARE IMPLEMENTATIONS FOR LOW-LEVEL VISION ALGORITHMS

Ed Gamble

Abstract: Early and intermediate vision algorithms, such as smoothing and discontinuity detection, are often implemented on general-purpose serial, and, more recently, parallel computers. The excessive time required by these general-purpose computers prevents real-time computation of these vision algorithms. Special-purpose hardware implementations of low-level vision algorithms may be needed to achieve real-time processing.

This memo reviews and analyzes some hardware implementations of low-level vision algorithms. Two types of hardware implementations are considered: the digital signal processing chips of Ruetz (and Broderson) and the analog VLSI circuits of Carver Mead. Both these approaches claim to achieve real-time image processing; both have limited the vision problem that they solved in ways largely inconsistent with vision processing in unrestricted environments. The advantages and disadvantages of these two approaches for producing a general, real-time vision system are considered. As early attempts at comprehensive vision hardware, these two approaches provide useful insights for future developments of vision hardware.

This paper describes research done within the Artificial Intelligence Laboratory. Support for the A.I. Laboratory's artificial intelligence research is provided by the Advanced Research Projects Agency of the Department of Defense under Army contract DACA76–85–C–0010, and in part by ONR contract N00014–85–K–0124.

1 Introduction

The purpose of this paper is to compare two approaches to special-purpose hardware for vision: the analog VLSI approach of Carver Mead[1] at Caltech and the digital VLSI approach championed by Ruetz[2] at Berkeley. These two researchers have adopted fundamentally different views on the implementation of vision algorithms in hardware. This paper will provide an overview of their techniques, assumptions, perceived motivation and philosophy. These issues have important consequences for future developments of vision hardware, including the recent M.I.T.[3] proposal.

The fundamental problem of machine vision is to recognize objects and to navigate through an environment by processing of camera images. This problem of machine vision is typically broken into three levels: early vision, intermediate vision, and recognition[4]. These three levels are all computationally intensive. Among these three levels, early and intermediate vision algorithms have similar computational requirements. Early and intermediate vision are charged with taking the input data, at camera rate, and producing a lower complexity, symbolic representation of the scene. The early vision level processes the input images to determine surface properties in the 3-dimensional scene. Typical surface properties are: depth, motion, color or albedo, and texture. The task of intermediate vision is to compute the discontinuities in the surface properties provided by early vision. The discontinuities mark abrupt changes in surface properties and usually correspond to object boundaries.

By comparison, the recognition level is computationally intensive because of the combinatorics of recognition. Recognition uses the object boundaries provided by intermediate vision to identify the objects. These object boundaries may be symbolic representations, a "feature," such as a line (modeled by, say, position, length, angle, and strength). For typical scene features, recognition database sizes, and model features, the possible combinations quickly become overwhelming. Recognition algorithms are drastically different than the generally pixel-based algorithms of early and intermediate vision. For this reason, this paper will not deal with the problems of hardware for recognition. Rather the focus will be on pixel-based algorithms for early and intermediate vision.

The two levels, early and intermediate vision, share some low-level vision

algorithms such as smoothing and discontinuity detection. For many years these algorithms were implemented on general-purpose serial, and, more recently, parallel computers. The use of a general-purpose computer facilitates modification to the algorithms as research objectives change. A draw-back of such computers has been the excessive time required for the computations. On a serial computer, even the relatively primitive operation of edge detection can take minutes or, on a parallel computer, seconds. Worse still, a sophisticated algorithm for smoothing surface property data while preserving discontinuities[5] can take minutes on a parallel computer such as the Connection Machine. Neither of these speeds approach the camera frame rate. The need for vision hardware derives from the difficult computational requirements during the early stages of vision processing due to the large data rate.

Both the approaches of Mead and Ruetz claim to perform real-time image processing. As discussed later, both have limited the vision problem that they solved in ways largely inconsistent with a vision system for general environments. The approach of Ruetz limits the vision problem to two-dimensional, motionless images with constraints on the image backgrounds. Mead's approach is limited in one regard by its photosensor resolution which mandates coarse image analysis and, consequently, is probably unusable for recognition tasks.

Although limited, these two approaches, as early attempts at comprehensive vision hardware, provide useful insights for future developments of vision hardware. For example, the trade-offs between local processing and photodetector density in Mead's approach must be addressed when contemplating vision hardware. Such decisions have important consequences in design time, circuit modularity, and optical properties.

The reasons for the differences in these two approaches to vision hardware stems largely from philosophical differences and goals. Mead appears to be more interested in using VLSI to explore biological implementations. His "silicon retina" is one example where the circuit design is driven by the biological design. Ruetz takes an engineering point of view in which a functioning, noiseless device is produced even if it poorly approximates the ultimate problem to be solved.

The remainder of this paper is organized into four sections. The first section supplies an overview of the vision problem and outlines various possible

algorithms for vision hardware. The second and third sections are devoted to the two hardware approaches. Each section details the vision problem solved, the background information regarding the hardware, and the advantages and limitations of the hardware as implemented. The philosophy and goals driving the research for these two approaches is also discussed. The final section provides a more direct comparison between the two approaches and includes suggestions for future developments as a convergence of techniques.

2 Vision Algorithm Primitives

In this analysis of vision algorithm primitives the emphasis is on the early and intermediate stages of vision processing. The primary outputs for this processing are the discontinuities in the surface properties and, to a lesser extent, the surface properties themselves. These outputs would be subsequently processed by a recognition system to identify objects in the 3-dimensional scene. This recognition process will not be discussed here.

2.1 Edge and Discontinuity Detection

Discontinuity detection is basically a generalization of the problem of edge detection. Figure 1 provides an overview of early vision and discontinuity detection. This figure shows the 3-D scene composed of M objects. All the points on each object have a surface property vector

$$X_i = \left\{ \begin{array}{l} \text{Position, } \vec{r} \\ \text{Texture Class} \\ \text{Velocity} \\ \text{Surface Color} \\ \cdots \end{array} \right\},$$

where X identifies the imaged point (i.e. pixel) and i is the object label. The 3-D scene is imaged by one or more optical systems, at repeated times, to yield a set of images, $I(x, y)$, distinguished by the time of measurement and the position of the optical system. The task of early vision is to use the set of images to determine this surface property vector at a subset of

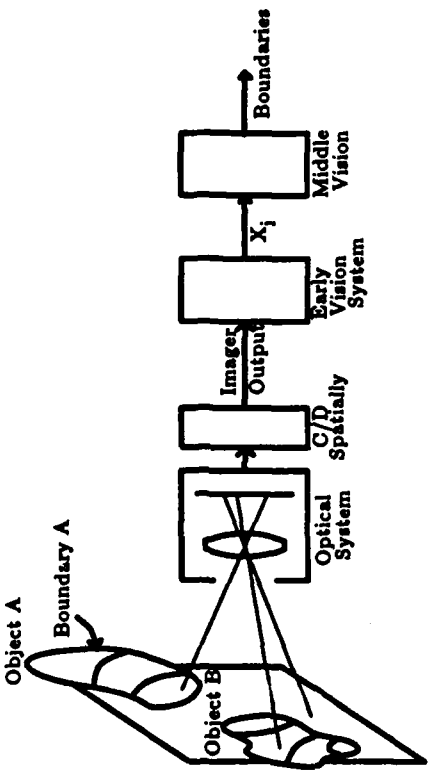

Figure 1: An overview of the early and intermediate vision tasks.

the image pixels. For example, a stereo algorithm produces the position, a motion algorithm determines the velocity, and a texture algorithm classifies the texture of a pixel. The surface property vector is then used by intermediate vision to ascertain the boundaries for each object i, ($i \in M$), and consequently to group the image points X_i comprising each of the i objects.

Imaging technology is such that the images, I, are spatially sampled by the imaging device. The imaging devices respond to incident light intensity and are typically arrays of photodetectors like charge-coupled-devices (CCDs) producing charge or phototransistors (PTs) producing current. The photodetector arrays can be linear or rectangular, hexagonal[1], or even "foveal"[6]. The output from the imaging device can be either continuous in time or discrete. CCDs are clocked devices that gather charge to produce a discrete-time signal. PTs produce a continuous-time signal. Both signals have analog magnitudes. Once the image has been detected by the imaging device, the signal processing begins at each pixel.

Edge detection entails finding those locations in the image where the incident light intensity varies rapidly in space. This is performed by finding the maximum in the gradient or the zeros in the second derivative. The problem of differentiation is difficult because of the presence of noise in the image signal. The noise is reduced by filtering or, equivalently, smoothing; however, smoothing has the undesirable characteristic of also reducing the edge signal.

2.1.1 Smoothing Techniques

One approach to edge detection is to convolve the image signal with a Gaussian and then to look for zeros in the Laplacian of the convolved output[7]. Convolution with a 2-D Gaussian has several convenient qualities[8]: 1) the kernel is circularly symmetric and therefore does not favor any direction *a priori*, 2) it is separable in x and y, 3) its Fourier transform is also a Gaussian, and 4) it can be approximated by a binomial series.

The binomial approximation to convolution with a Gaussian is made by repeatedly convolving the image with the mask $\{1/2, 1/2\}$. Performing a binomial convolution has several favorable attributes for hardware implementation. First, only local pixel access is required and second, scaling is

by factors of 2 which is more easily implemented in some hardware systems. Each convolution with the $\{1/2, 1/2\}$ mask yields successive terms in the binomial series.

In practice, either type of convolution, Gaussian or its binomial approximation, is acceptable. Of course Gaussian convolution is much simpler to analyze theoretically; however, the accuracy of edges should not "make or break" a vision system (at least until proven otherwise). This results from the considerable confusion regarding what edges are optimal for recognition.

A more general approach to smoothing that proves useful for hardware implementations is regularization[9]. The regularization formulation for vision seeks to minimize the error between the input signal and the output signal subject to constraints. The constraints are designed to impose *a priori* assumptions about the nature of the solution. For example, with an input signal of light intensity and a smooth output signal desired, an appropriate constraint might be the gradient of the output. The following equation expresses this notion for a continuous output field, $f(x)$ given input $g(x)$.

$$E = \int \left[\alpha(f-g)^2 + \lambda\|\nabla f\|^2\right] dx \tag{1}$$

The function, $f(x)$, that minimizes E is sought. The first term in this equation requires that f be close to the input data g; the second term requires that f be smooth.

Finding the minimum of Equation 1 is a problem in variational calculus. The solution for $f(x)$ is found by solving

$$-\lambda\nabla^2 f + \alpha f = \alpha g \tag{2}$$

For a 1-D problem with continuous data, the Fourier transform of Equation 2 is

$$F(\eta) = \frac{\alpha}{\alpha + \lambda\eta^2} G(\eta), \tag{3}$$

where η is the angular spatial frequency. The regularized solution can be viewed as nothing more than a convolution of the input with a low-pass filter.

Formulating the vision problem as an energy minimization is natural for implementing the problem in a physical system[10, 11]. Physical systems, electrical, mechanical, etc, minimize the system's Lagrangian. For the case

of an electrical network the Lagrangian is simply the network's energy. If the electrical network's energy is designed to duplicate a vision problem's energy, the network will solve the vision problem.

2.1.2 Edge Detection

Many edge detection techniques exist. Possibly the most studied and most biologically relevant edge detector may be Gaussian convolution followed by application of the Laplacian operator[7]. The computation of the Laplacian of a Gaussian (LOG) convolution can be performed or approximated in several ways. One way is to first convolve with the Gaussian and to subsequently compute the Laplacian with one of its masks[12]. This is natural for many digital systems. An approximation to the LOG is based on the biological "center-surround" receptor and entails simply subtracting the smoothed background, the surround, from the signal, the center. After the LOG calculation, edges are identified by the zero-crossings. Yet a third way, based on the approximation to the LOG as a difference of Gaussians (DOG), is to convolve with two Gaussians and then to subtract the two.

The DOG approximation to LOG convolution has been implemented in hardware[13]. The implementation exploits two observations: 1) the solution to the diffusion equation is the convolution of the initial distribution with a Gaussian and 2) the voltages in a distributed resistive/capacitive transmission line obey the diffusion equation. The width of the Gaussian is a function of time so that the DOG is computed by sampling the voltages on the transmission line at two times and then subtracting them.

Discontinuity detection can be viewed as generalized edge detection. Edge detection seeks discontinuities in the light intensity; discontinuity detection seeks discontinuities in surface property data. Using the surface property data, such as depth from stereo, adds an additional complication since the data can be sparse. The surface property data is sparse because some early vision algorithms produce surface property data only at intensity edges.

2.1.3 Smoothing with Discontinuity Detection

One problem with the preceding edge detector analysis is that the smoothing process reduces precisely that signal from the differential operator needed to identify the edge. The edges themselves are smoothed away. To some extent this problem can be eliminated by combining the smoothing and edge detection processes[14, 11, 15, 16, 17]. These techniques smooth the data unless a discontinuity is detected. Smoothing is abandoned between locations separated by a discontinuity. The output is the smoothed data and the discontinuities in the data. The computation proceeds by finding the configuration of data and discontinuities that minimizes a function. An example function is shown below.

$$E_i = \sum_{j \in C_i} \{(f_i - f_j)^2(1 - l_{ij}) + \alpha(f_i - g_i)^2 + \beta V_C(l_{ij})\} \qquad (4)$$

The variable f_i is the output data at site i; l_{ij} is the output discontinuities, a binary value, separating site i and j. The function is designed to impose constraints of smoothness and continuity on the output data and discontinuities. The function E_i is not quadratic and, consequently, stochastic methods must be used to minimize E_i.

3 Analog VLSI

This chapter describes the use of analog VLSI devices for vision work. Largely initiated by Caltech's Carver Mead[1], analog VLSI is now also used by, among others, Christof Koch[18], also at Caltech. Another approach to analog computation of vision algorithms utilizes CCD technology[19, 20, 21]. However, within the scope of this paper, the CCD technology will not be analyzed.

This chapter is divided into three sections. The first section contains a discussion of subthreshold CMOS devices and is followed by a section on hardware implementations of vision algorithms. The final section analyzes the analog VLSI's applicability for implementing a real-time vision system.

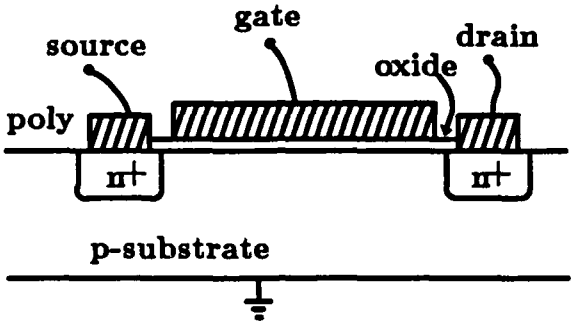

Figure 2: An n-channel MOS device. P-channel devices are fabricated within an n-well. The device parameters are presented in Section 3.1.2.

3.1 Subthreshold CMOS

The use of MOS devices in the subthreshold regime has been championed by Carver Mead[1]. For vision applications, the subthreshold regime is preferred by Mead for three reasons: 1) the exponential dependence of the drain current as a function of gate voltage, 2) the low power usage in this regime, and 3) the near current source characteristic of the source-drain terminals for $V_{ds} \geq\sim 100mV$. The following sections describe the basic device physics of subthreshold MOS operation, outline the circuit model, and presents some of the limitations and advantages of these CMOS devices.

3.1.1 Device Physics

The majority of MOS devices are usually not operated in the subthreshold region. Most texts, in fact, call the drain current zero unless the gate voltage is above the threshold voltage, while for gate voltages above this threshold the drain current is linear or quadratic in the gate voltage. Figure 2 shows an n-channel MOS structure and will be used during the description of MOS operation.

When a positive voltage is applied to the gate, a "channel" forms just below the gate oxide in the p substrate. The channel is formed by expelling the majority carrier holes which leaves a depletion region of fixed acceptor

atoms. If the gate voltage is large enough, free-moving, minority-carrier electrons can also occupy this depletion region. Both the fixed acceptor atoms and the induced, free electrons within this depletion region balance the positive charge on the gate electrode; the gate oxide acts like a capacitor.

When the density of free electrons in the depletion region number much less than the acceptor ion density, the MOS is in the subthreshold region. If these free electrons are ignored and Gauss's law is applied to the oxide/substrate interface, there can be no electric field parallel to the surface and, therefore, the interface is an equipotential surface. Any free electron motion along the interface cannot be due to drift, only diffusion. To compute this diffusion current the source and drain voltages must be considered.

The current due to diffusion is given by

$$I = \frac{wqD}{l}(N_{dg} - N_{sg}) \tag{5}$$

where q = electronic charge, w = transistor width, D = diffusion constant, N_{dg} = electron density at the drain-gate region, and N_{sg} = electron density at the source-gate region. l is the length of the channel. The gate surface potential and the electron density of states provides the means to compute the electron densities. The density of states for electrons is a Fermi distribution but, far away from the Fermi energy, the distribution can be approximated by a Boltzman distribution. The resulting electron density is

$$N = N_0 e^{q\psi/kT} \tag{6}$$

where ψ is the gate surface potential. Combining this with Equation 5 the drain current is

$$I = I_0 e^{\kappa V_g/\beta} e^{-V_s/\beta}(1 - e^{-V_{ds}/\beta}) \tag{7}$$

where $\beta = kT/q = 25mV$ and

$$I_0 = \frac{wq}{l} D N_0 e^{-\phi_0/kt}.$$

For small V_{ds}, the channel acts like a linear resistor. As V_{ds} increases the channel gets pinched off near the drain. Further increases in V_{ds} pinch off the channel completely and cause the channel to separate from the drain. This separation reduces the effective length of the channel below l. With the channel pinched off, variations in V_{ds} do not effect the electron density; the

channel current becomes largely independent of V_{ds}. The channel length does depend on V_{ds} and this small dependence affects the drain current and accounts for the slight slope in the I-V characteristics in the saturation region.

3.1.2 MOS Specifications and Limitations

The analog VLSI circuits are produced by the MOSIS foundry using 2 μm technology. Most publications for vision applications of analog VLSI do not reveal specific numbers characterizing the performance of these devices. However, some data can be found[1] or inferred from the MOSIS design specifications. The channel length l is about 1.5 μm; oxide thickness is 125$\mathring{A}$. For a silicon dioxide gate, t' . capacitance is about $0.1pF$. The factor, κ, may range from 0.55 to 0.73 with 0.7 being a typical value. The current I_0 is approximately $1.5\text{x}10^{-7}$ nA. Gate voltages, V_{gs}, for subthreshold operation are generally between 0.3 and 0.8 volts with the corresponding drain currents of $7\text{x}10^{-4}$ and $8\text{x}10^{2}$ nA respectively. (Note that some of the numbers may seem inconsistent. I_0 was deduced from a device with $\kappa = 0.676$ ([1], page 38) but the drain currents were computed witn $\kappa = 0.7$.) Normal threshold for the MOS device is roughly $V_{gs} \geq 1$ Volt. The device behaves similar to a current source when in the *saturated* region for $V_{ds} > 100$ mV.

The primary limitation of MOS devices operating in the subthreshold region arises from the inability to provide a consistent threshold voltage across the chip die. This is the problem of device *mismatch*. The threshold voltage is part of ϕ_0 in the previous section. Because of the exponential dependence, small variations of ϕ_0 can introduce large variations in I_0. For instance, if $\phi_0 = qV_0$ and $V_0 = 10mV$, the variation in I_0 is nearly 33%. For transistors that are physically close to one another, a typical variation is $\pm 20\%$ [1] although variations of 100% may in fact be more representative[22] for device mismatch.

Computations that use differential amplifiers, such as derivatives, are sensitive to variations in I_0. Small differential signals may be overwhelmed by the transistor mismatch and, consequently, the circuit design must minimize this effect. Of course biological systems have device mismatch and these systems generally do fine. As a scientific endeavor, the mismatch is acceptable; as an engineering endeavor, the mismatch is problematic and

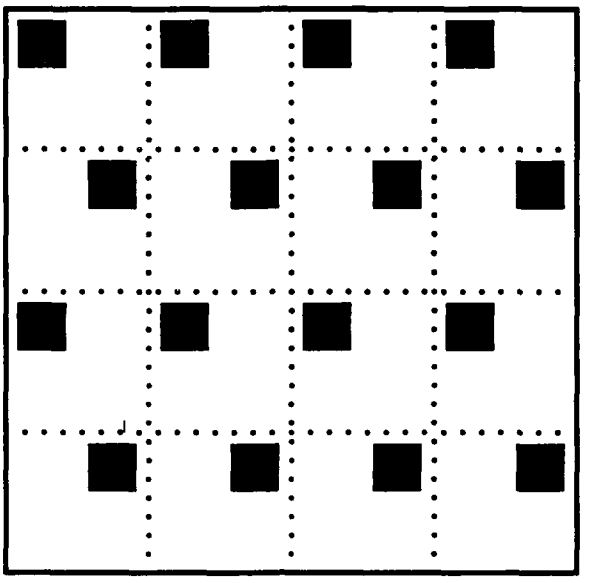

Figure 3: Hexagonal lattice of pixels. Each pixel contains a photodetector of area a, the black square, surrounded by local processing circuits. Each pixel has an area of A ($A > a$).

hampers development of a useful vision system.

3.2 MOS Vision Algorithms and Devices

In this section two of the higher-level analog VLSI circuits will be presented. These circuits are the silicon retina and the resistive fuses. These circuits utilize some common elements, such as the phototransistor and the resistive network for smoothing, and a common layout structure. These shared structures are discussed first as background for the subsequent discussion of the two higher-level circuits.

3.2.1 Common VLSI Structures

Figure 3 shows the typical layout for the analog VLSI circuits. Each

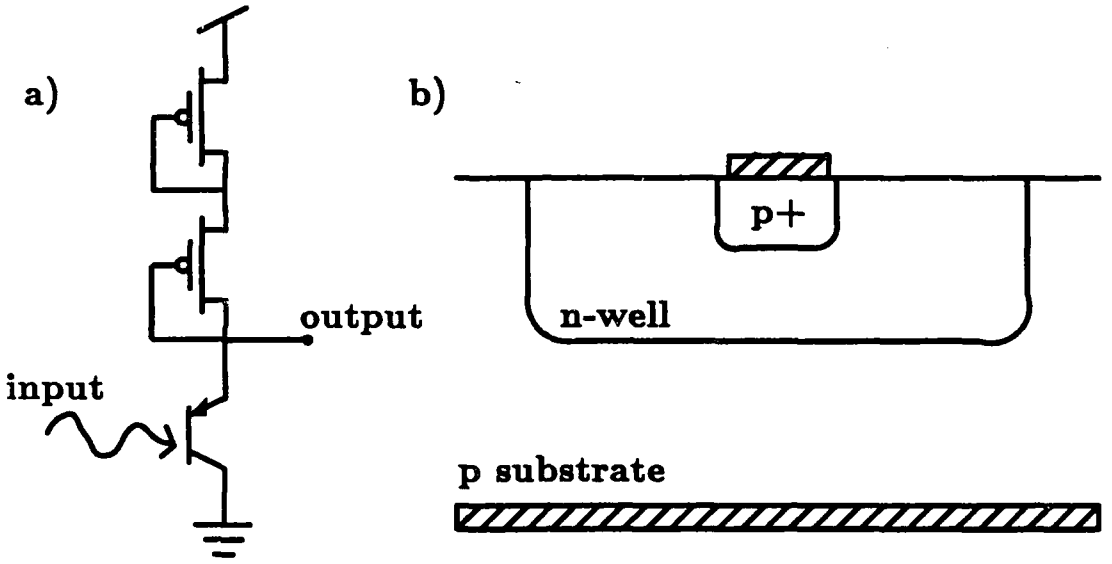

Figure 4: a) The phototransistor circuit[1]. b) A pnp phototransistor device fabricated with n-well CMOS.

circuit consists of a one or two dimensional grid of pixels. The inter-pixel spacing is defined as L; the pixel area as $A(= L^2)$. Each pixel is comprised of a phototransistor and additional local-processing circuitry. Each phototransistor has size l and area $a(= l^2)$. The area-fill-factor is η_A and is defined as a/A. The number of pixels along each linear dimension is N. The two dimensional grid is arranged as a hexagonal lattice by displacing alternate rows by $L/2$.

The phototransistor circuit and device are shown in Figure 4. The pnp transistor has a photosensitive base region which produces a current at the collector that is proportional to the incident light intensity. This photocurrent is fed through one (or two) diode-connected p-channel MOS device. The output voltage for this circuit is proportional to the logarithm of the photocurrent,

$$V_{output} = V_{dd} - \frac{2\beta}{\kappa} ln(\frac{I}{I_0}),$$

where I is the current through the phototransistor's collector. The loga-

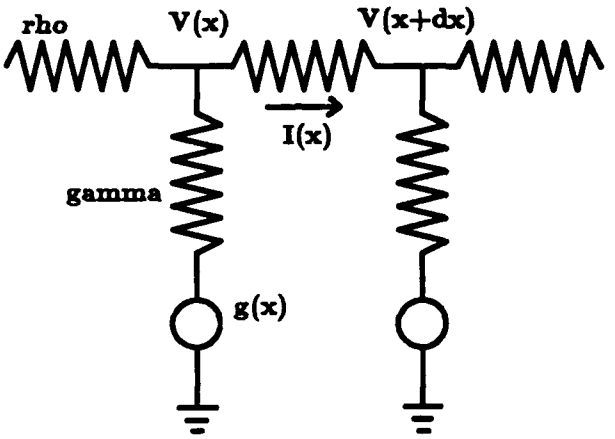

Figure 5: A resistive network.

rithmic compression increases the usable range of incident light intensities. Typically V_{output} is 1 to 2.5 Volts below V_{dd}. This corresponds to a current range of roughly 5 orders of magnitude. The smallest detectable current is about 10^{-5} nA or 10^5 photons/second.

With n-well CMOS, for the pnp phototransistor, the base is the well itself and the emitter is fabricated from a p-diffusion step. The p-type collector is the substrate (Figure 4b) and it is electrically grounded. The n-well process produces parasitic phototransistors whenever a well is deposited. To avoid unwanted photocurrents, the die is shielded everywhere except at the desired phototransistors. The second metal layer serves as the shield.

Figure 5 shows the third and final common structure of this section: the resistive network[1]. The resistive network performs a smoothing operation useful for early vision and is used in both the silicon retina and the discontinuity-detecting resistive fuses. The one-dimensional, continuous resistive network solves for the minimum of Equation 1. With a resistivity per unit length of ρ, a conductivity per unit length to ground of γ, and an input voltage of $g(x)$, Kirchoff's current and voltage laws are:

$$\frac{dI(x)}{dx} = \gamma[V(x) - g(x)]$$

$$\frac{dV(x)}{dx} = \rho I(x) \tag{8}$$

These two equations yield Equation 2 provided $\lambda = 1$ and $G = \rho\gamma$. The Green's function is

$$V(x|x_0) = \begin{cases} Le^{-(x-x_0)/L} & x \geq x_0 \\ Le^{(x-x_0)/L} & x < x_0 \end{cases} \tag{9}$$

where the space constant (or "smoothing width") is $L = 1/\sqrt{\gamma\rho}$. Equation 9 shows that a unit impulse at $x = x_0$ diffuses throughout the network with an exponential fall-off. If a capacitance is added between $V(x)$ and ground, the network converges to a solution with a time constant of roughly $\tau = C/G$.

Figure 6 shows an implementation of a resistor and its adjacent nodes for the resistive network[1]. The resistor is comprised of the transistors labeled $Q1$ and $Q2$ in Figure 6b. To find the small-signal resistance between V_n and V_{n+1}, assume that $\kappa V_{gm} - V_m = V_d$ for $m \in \{n, n+1\}$. Transistors $Q1$ and $Q2$ of Figure 6b will be biased identically and the resulting current will be

$$I = I_0 e^{V_d/\beta} \tanh[\frac{(V_n - V_{n+1})}{2\beta}]. \tag{10}$$

For small signals ($x \leq 0.2$), $\tanh(x) = x$ and the resistance is

$$R = \frac{2\beta}{I_0 e^{V_d/\beta}}.$$

The resistance can be modified by varying the voltage V_d. This voltage V_d is the gate-to-source voltage of Q_d shown in Figure 6a. For this circuit, the current mirror, $Q3 - Q4$, keeps the currents through $Q1$, $Q2$, and Qd all equal to $I_b/2$. The diode connection at $Q2$ has voltage V_n and consequently

$$\frac{I_b}{2} = I_0 e^{(\kappa V_{gn} - V_n)/\beta} = I_0 e^{V_d/\beta}.$$

Or, in terms of the bias voltage V_b, the resistive network's resistance is

$$R = \frac{4\beta}{I_0 e^{\kappa V_b/\beta}}.$$

This analysis assumes that all the transistors are well matched and operating in the saturation region. The measured I-V characteristic[1] for the horizontal resistor shows that the small-signal assumption is valid for roughly

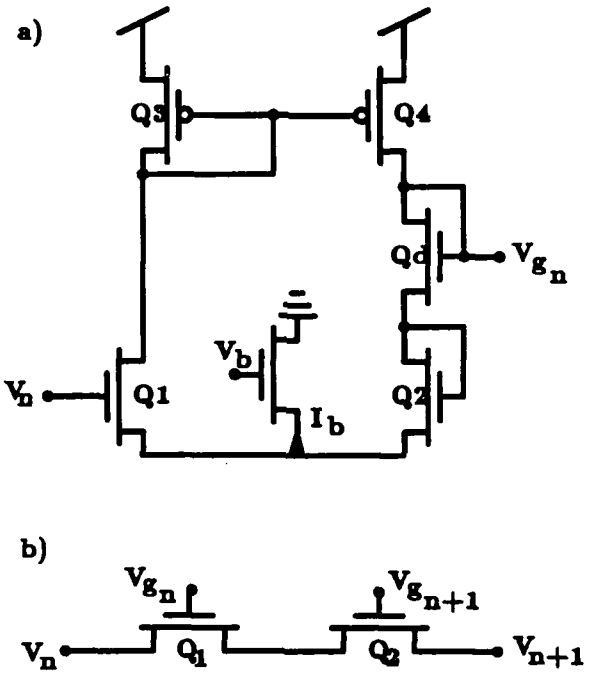

Figure 6: The resistive network[1]. a) The bias circuit for the nth network node. V_n is connected to a node in the network and V_{g_n} is connected to all the transistors adjacent to the node. For a hexagonal lattice V_{g_n} is attached to 6 gates. b) The transistors Q1 and Q2 model a resistor between nodes n and $n+1$.

$V_n - V_{n+1} = 100mV$ with $V_{n+1} = 2.5V$. Within this 100 mV range, the resistance is linear and ranges from between about $0.2M\Omega$ and $2\text{x}10^4 M\Omega$ (for $0.4V < V_b < 0.8V$).

Equation 10 shows that the current between nodes V_1 and V_2 of Figure 6a saturates when $|V_1 - V_2| \gg 0.0$. This is a crude type of discontinuity detector. At saturation, the current is $I = I_0 e^{\beta V_d}$ and the effective resistance approaches ∞. The two voltages across the resistor are no longer related; smoothing no longer occurs. A more sophisticated discontinuity detector is discussed in the subsequent section on resistive fuses.

3.2.2 Silicon Retina

The retina is the first stage of the image processing that ultimately converts the image produced by the eye's optical system into moving, colorful, recognizable objects. The optical signal is converted to an electrical signal by the photoreceptors that line the back side of the retina. Subsequent layers of retinal cells: amacrine, horizontal, bipolar and ganglion, further process the electrical signal until the ganglion axons send the signal along to the lateral geniculate nucleus. Presumably, the different cell types are associated with different computations. Some of the ganglion cells may produce something similar to the convolution of a Laplacian of a Gaussian. The horizontal cells produce the surround region and the bipolar cells produce the center region. The ganglion cells produce a center-surround response by subtracting the bipolar and horizontal cell outputs. The amacrine cells respond to time-varying signals. The resulting computation yields those regions in the image that change spatially or temporally.

The silicon retina[23] is an attempt to duplicate, at Marr's computational and algorithmic levels, the simplified retina described above. A phototransistor imitates, in an approximate way, the human retina's photoreceptor response. A center-surround algorithm is used by the silicon retina to find spatially varying regions. The horizontal resistive network provides the surround region and a differential amplifier subtracts this from the photoreceptor response. The resulting signal is clocked off the retina for display.

Figure 7 provides an overview of the silicon retina circuitry. Most of the elements have been described previously. The phototransistor circuit

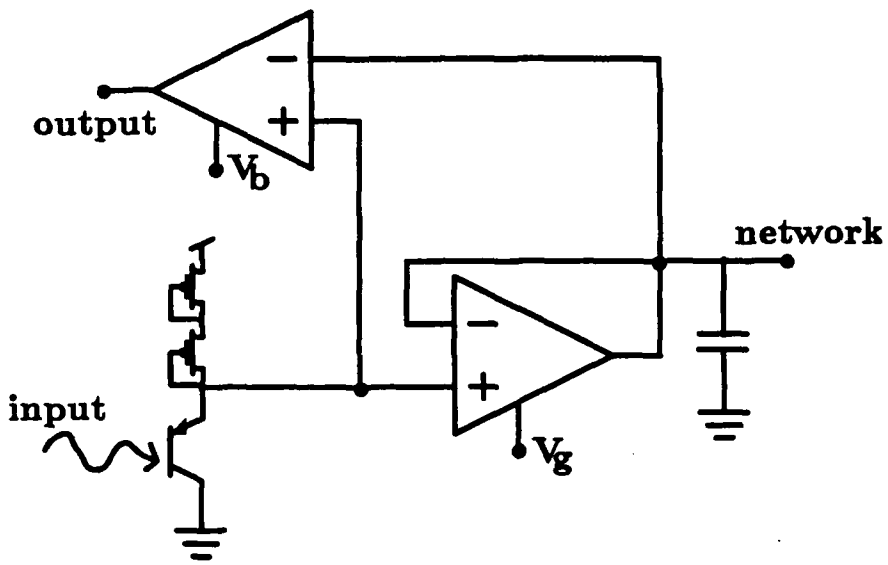

Figure 7: The silicon retina[1]. The output is the difference between the input voltage and the smoothed output of the resistive network. The two differential amplifiers shown (biased by V_b and V_g) are transconductance amplifiers [1].

provides a voltage for the follower-connected differential amplifier. This amplifier acts like a conductance to couple the phototransistor voltage to the resistive network node. The conductance is determined by voltage V_g and has a range similar to the reciprocal of the horizontal resistance. The resistive network couples the different pixels by a resistance determined by V_R of Figure 5.

Several aspects of the silicon retina are variable. The smoothing width is determined by L of Equation 9 which, for the exponential dependence of R and G on gate voltage, is controlled by the difference of voltages V_g (Figure 7) and V_R (Figure 6). Typically L ranges between 0.1 and 10.0 pixels. The time response τ is controlled by G and thus V_g as well as the fixed capacitance C. The capacitance, as mentioned in Section 3.1.2, is about 0.1 pF. Given the range of G the network's time response can be varied between about 1ms and 10ns or 1kHz to 100 MHz (other capacitance probably makes this an unrealizable speed). The smoothing width and time response are independently variable.

The differential amplifier at the output computes the difference between the phototransistor voltage and the smoothed resistive network voltage. This is the center-surround computation. The output from the amplifier is enabled by the voltage V_b.

The silicon retina contains 48 x 48 pixels. Each pixel is comprised of the phototransistor, the circuitry shown in Figure 7, and the resistive network. A pixel is roughly 100 x 100 μm^2 and the phototransistor occupies 10 % of the pixel area. Besides the circuitry for each pixel, the silicon retina contains devices to access each pixel's output current. Any one pixel's response can be observed over time or each pixel can be sequentially clocked out for video display. A single pixel's intensity, time, and edge response is qualitatively similar to measurements made on biological retinas[23]. When each pixel is sequentially clocked out, each pixel should reach equilibrium at the 30 Hz frame rate. If the retina is scaled from 48 x 48 to 512 x 512, the speed for each pixel's computation, as determined primarily by the time response of the resistive network, need not increase. Only the sequential clocking circuitry must be sped up.

3.2.3 Resistive Fuses

The use of resistive fuses[24] attempts to implement a solution to important problem of discontinuity detection[14]. The function of a fuse is to prevent the smoothing between neighboring sites. Once broken, the fuse marks the location of the discontinuity and prevents further smoothing between the neighboring sites. The smoothing is avoided by greatly increasing the resistance between the sites.

Equation 1, embodying the smoothing problem, is modified by the addition of discontinuities. In discrete form the energy for smoothing with discontinuities is:

$$E_i = \sum_{j \in C_i} \{(f_i - f_j)^2(1 - l_{ij}) + \alpha(f_i - g_i)^2 + \beta l_{ij}\} \tag{11}$$

Here g is the surface property data, an input; f and l are the smoothed surface property data and the discontinuities respectively, the outputs. The total energy is the sum of E_i for all sites i. The field l is a binary field so that when $l_{ij} = 1$ the first term in Equation 11 contributes nothing to the energy and the third term contributes β. β is the penalty for turning on a line. As a function of $f_i - f_j$, the minimum of E_i is quadratic with $l_{ij} = 0$ until $(f_i - f_j)^2 = \beta$ where $l_{ij} = 1$ and $E_i = \beta$. A similar dependence can be implemented in analog VLSI.

The previous analysis showed that when $\Delta f \geq \sqrt{\beta}$ the energy is constant; prior to that point, the energy is quadratic. A fuse has just that property. In analog VLSI, a fuse is implemented by making the voltage V_b of Figure 6b a function of the voltage difference between nodes in the resistive network. When the voltage difference is larger than some threshold, for an ideal fuse $R = \infty$ and V_b should be $0V$. An approximation to this has been implemented[24] and is shown in Figure 8. For this circuit, the fuse current is

$$I_{fuse} = \frac{1}{2}\left[I_B - I_A \tanh\left(\frac{\kappa|\Delta V|}{2\beta}\right)\right] \tanh\left(\frac{\Delta V}{2\beta}\right). \tag{12}$$

The current I_B determines the resistance for smoothing; the current I_A determines when the resistance breaks (really, begins to break). Both are adjustable but not separately for each node in the circuit.

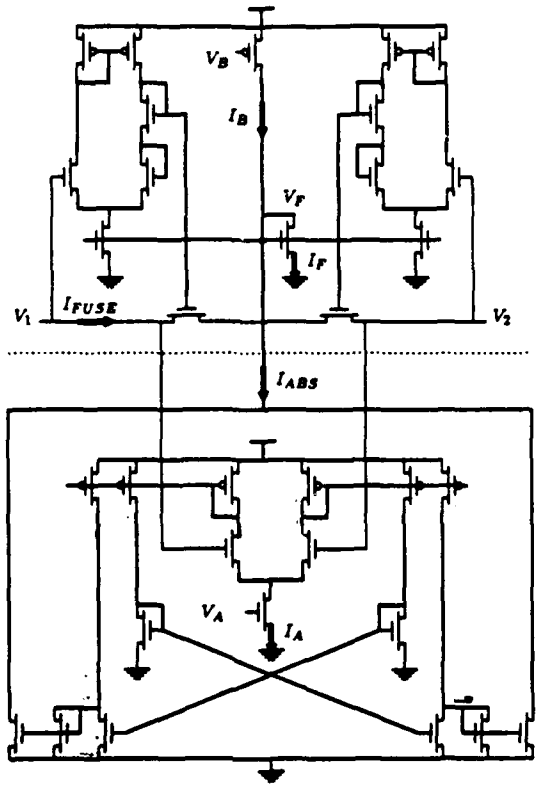

Figure 8: A resistive fuse implementation[24].

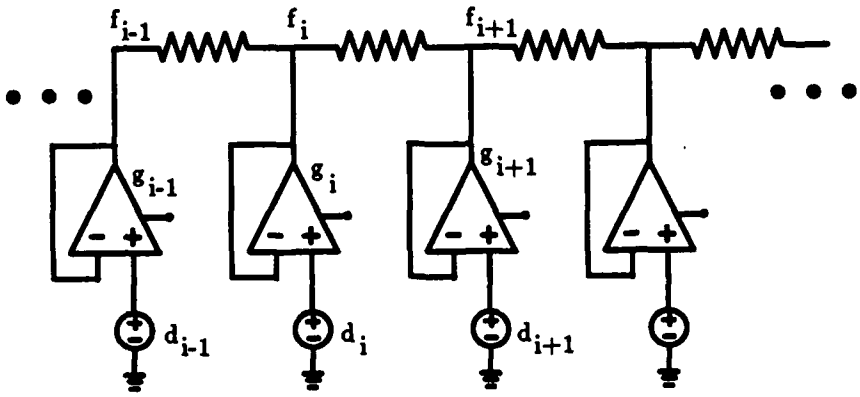

Figure 9: A resistive fuse network[24].

These fuses have been used in an eight node network. Figure 9 is a block diagram for the network. Each of the eight input voltages d_i are variable and each smoothed output voltage f_i is accessible. The conductances g_i are analogous to α of Equation 11 and are a measure of the expected noise or "trust" of the input data d_i. Each of the g_i are variable and, when the d_i are sparse, some of the g_i will be zero. The currents I_B and I_A are controlled by voltages V_B and V_A, respectively. The network has been shown to smooth data and break the smoothing to mark discontinuities[24]. A two-dimensional circuit with 400 nodes is in development.

3.3 Discussion

One motivation driving Mead's work appears to be the desire to study biological systems by building analogous systems in hardware. Building these hardware systems serves two complementary roles[1](page 8). They attempt to provide computational neuroscientists with a facility which allows experimental verification of the neuroscientist's hypotheses. Additionally, development of these hardware systems attempt to provide insight into the properties of collective systems. These are the main issues guiding the research on analog VLSI.

Yet, from a computational neuroscientist's view, hardware systems do not provide the required flexibility for algorithm development. So far, the design of the hardware has been guided by the results from the computational neuroscientists; not the other way around. The hardware implementations are approximations to the computational theory. Discrepancies between hardware results and theory reveal the inadequacy of the hardware implementation and representation. The discrepancies have not been attributed to the computational theory. The analog hardware does not seem to have satisfied the goal of providing neuroscientists with an experimental facility.

As more tools, techniques and experience develops, analog VLSI may eventually contribute to the computational theory. These developments, expressed as a set of VLSI "standard cells" or modules, may allow the hardware designer to rapidly modify an algorithm thereby reducing development time. The resistive network may be an example of an emerging standardized module. Another impediment to analog VLSI's contribution may be price. Until such a time that these impediments are circumvented, the primary

tool of computational neuroscientists will remain software simulation.

These impediments are not the major factors limiting the use of analog VLSI in a general vision system. The fundamental problems that ultimately limit analog VLSI's usefulness in vision are discussed in Section 3.3.3.

The biological focus of analog VLSI systems hinders engineering of a robust vision system. A biological model may demand local processing with three-dimensional circuitry in nature and, currently, two-dimensional circuitry on silicon. However, two-dimensional local processing has horrible scaling properties as computational requirements and pixel densities increase. Another hindrance, based on biologically acceptable power consumption, is the use of subthreshold MOS devices. In the subthreshold regime, owing to the exponential dependence of drain current on gate voltage, MOS devices are more difficult to manufacture with uniform properties. Consequently noise issues must be confronted. These examples of engineering difficiencies are discussed in more detail below.

3.3.1 Adaptive Retina

Although framed as a need to adapt the silicon retina to different light levels, the adaptive retina[25] is really an attempt to eliminate the problems of differential offset in the subthreshold MOS devices[26]. The mismatch between MOS device parameters proves particularly disruptive when computing derivatives. The simple differential amplifier can have a current-mirror with currents differing by 100% [1] and a 20% difference is common. Such differences can easily confound the center-surround computation of the silicon retina.

Figure 10 is a schematic of the adaptive retina. The adaptation serves to counteract the effect of mismatched devices. When the phototransistor emitter and the floating-gate[27] are exposed to UV radiation, the adaptive retina chip is illuminated with a uniform light intensity and the resistive network is set to compute a global average. Under these conditions, the UV radiation allows a small current to flow through the silicon-dioxide insulator between the floating-gate and the phototransistor's emitter. This current charges the floating-gate so as to reduce the surface potential of the p-channel within the floating-gate MOS transistor. Once equilibrium is

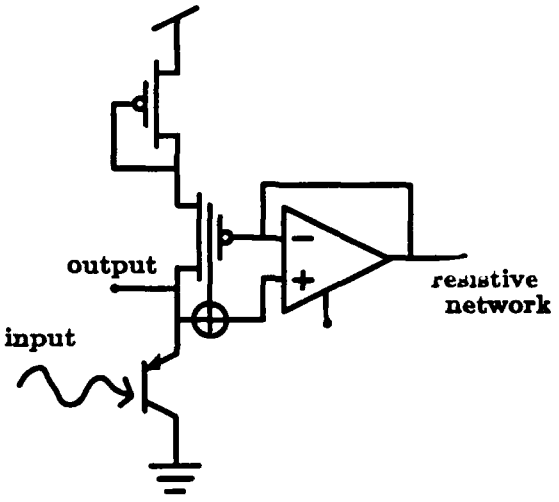

Figure 10: The adaptive retina[25]. While illuminated with ultraviolet light during adaptation, the $\oplus$ connects the adaptive retina output to the floating-gate[27]. Once the floating-gate charges, the adaptation is complete and the UV light is removed.

reached, V_{output} will be near the node voltage of the resistive network and hence equal for all pixels.

The adaptive retina turns out to exhibit properties similar to biological systems. Similar to biological retinas, the silicon retina can adapt to different light levels and also display "after-image" phenomenon[25]. One cannot argue with the results. Not unexpectedly, a biological approach reproduced biological results. Such results do not necessarily bring a working vision system nearer to reality.

3.3.2 Practical Resistive Fuse

For an analog VLSI circuit such as the resistive fuse, the primary goal has, once again, not been to develop a working vision system. Seemingly, and for good reason, the focus has been on understanding vision and vision algorithms. Some of the unanswered questions in vision and, particularly, intermediate vision are of a fundamental nature. In the case of the discontinuity-detecting resistive fuses, questions regarding parameter specification are exceedingly difficult and remain the major impediment to further development. The resistive-fuse system works under supervision when the parameters can be controlled; however, unsupervised, success is unlikely until parameter estimation issues are resolved.

The primary benefit resulting from resistive fuse hardware may be the speed which might allow exploration of the parameter space. Yet, in parameter estimation, the need to quickly modify the algorithm may show such a hardware approach to be ill-suited. The advent of a chip integrating intensity edges with surface property data tc detect discontinuities[26] may achieve more success When using intensity edges to guide the search for discontinuities in surface properties, the specification of parameters is significantly less critical[5, 28].

3.3.3 A Vision System in Analog VLSI?

Most likely, general working vision system in hardware will require two attributes: lots of pixels and lots of computation. These two attributes are lacking in the present analog VLSI implementations and are addressed by

the issue of *scaling*. Scaling refers primarily to increasing the total number of pixels and increasing the processing associated with each pixel. Large numbers of pixels are required to do anything more than just crude recognition and navigation and increasing the processing power is necessary when edge detection, stereo, motion, and discontinuity detection all must be performed.

As shown in Figure 3, analog VLSI technology positions the processing locally with each pixel. As the amount of required processing increases, the fractional area, η_A occupied by the phototransistor diminishes and, for the same number of pixels, the chip die size increases. Optical resolution and efficiency are both degraded when the local processing circuitry increases. Already, η_A is significantly smaller than current CCD technology utilizes. For the silicon retina, $\eta_A = 0.1$ (roughly); for the resistive fuses, η_A is even less. Both of these analog VLSI systems are very low level. Once circuitry for stereo and motion are added, as well as processing needed by intermediate vision, the optical performance may be reduced to unacceptable levels.

An analog VLSI layout designed to model more than one early vision module with two-dimensional local-processing would be highly non-modular. As currently formulated in computational vision theory, each of the individual vision modules requires the pixels to be locally interconnected. This interconnectivity is designed to impose the smoothness constraint on surface property data. With several vision modules implemented at each pixel, the interconnection layout may be prohibitedly complicated. A modular approach would have a chip (or seperate wafer region) for each vision module. Phototransistor or silicon retina output could be shared by all the chips.

When the number of pixels is increased (and/or the local processing requirements increase), designers of analog VLSI hardware must address the issues of wafer scale integration and, consequently, fault tolerant design. Biological systems have largely resolved both these issues. However, in circuit design, these issues are far from resolved and consequently vision hardware with analog VLSI must await further developments. In addition, wafer scale integration further increases the design time and device cost.

3.3.4 Review

The previous sections have detailed several of the disadvantages and successes of analog VLSI for vision. The silicon retina has been successful in duplicating, qualitatively, many of the characteristics of biological retinas. The adaptive retina successfully addressed the problem of MOS mismatch in the silicon retina and yielded "after image" effects similar to biological systems. At the computational theory level, both these devices as well as the resistive fuse and the stereo correlator[29] produced results consistent with theory. These systems used low-power, subthreshold MOS devices almost exclusively.

On the negative side, several problems with both analog VLSI and the design methodology for analog VLSI will hinder development of a vision system. In the subthreshold regime, MOS devices are difficult to match and, consequently, they demand robust circuit design. Redesign of analog VLSI circuits is difficult, because, due to its newness, analog VLSI does not have a standard set of circuit modules. With time both these problems may be reduced.

A significant problem with analog VLSI systems is the adherence to a local processing layout. As the local computation requirements increase, the optical resolution and response are reduced since the phototransistors occupy a smaller fraction of the pixel and are spaced further apart. Also, when additional vision modules are implemented in hardware, the local processing requirement reduces the modularity of the system. Finally, as the number of pixels increases, the circuits demand a larger portion of the silicon wafer. With larger wafer size, point defects will increase and fault tolerant circuits must be designed. These factors increase the cost, complexity and development time of analog VLSI systems.

4 Digital Circuits for Vision

General-purpose, serial and parallel, digital computers are used to implement vision algorithms. Because they are general-purpose computers, much of the hardware in these machines is not related to the specifics of vision tasks. The result is a computer with lots of flexibility and very little speed.

Although a research environment may not need real-time processing capability, most applications could benefit from vision algorithms running in real-time.

The processing required for real-time vision computations is immense. For a 512 by 512 image at 30 Hz the serial processing rate is nearly 10 MHz. This magnitude processing rate cannot be performed on a serial computer designed for general-purpose use. Even for a massively-parallel Connection Machine operating simultaneously on every element of the image, achieving a 30 Hz rate is difficult. Such a rate may be obtained on the Connection Machine for a fully configured (64k processors) machine, running an assembly-language coded version of an edge detector. Using a Connection Machine whenever a real-time vision task is required is beyond ridiculous.

Specialized hardware for vision may enable real-time computation. The previous section detailed the use of analog techniques for vision. This section examines an example of digital techniques for vision[2]. This section reviews the previous work, its goals and philosophy, and presents some of its algorithms and circuits. A discussion of the 3x3 convolver chip is detailed as well as the problems and inadequacies of the digital approach to vision.

4.1 Goals and Approach

The development of the image processing IC system[30] was guided by four major goals. From the standpoint of a vision researcher, the most important goal was that the system be able to perform image recognition on two-dimensional images. Another goal was that the system operate in real-time so that additional hardware of frame buffers would not be required. In addition, the design of the vision system should utilize modules that perform fundamental vision algorithms. In this way, the modules can be readily configured to solve different problems. The final goal was that development time be minimal.

In order to satisfy these goals, a strict hierarchy for the hardware design was employed. Each level in the hierarchy would contain those circuits required by one or more modules of a higher level. In this way duplicate design could be eliminated provided that the circuits could be generalized. Design at a higher level would entail primarily interfacing the "building-

block" modules from the lower level. This design hierarchy also serves to reduce development time by utilizing identical modules at each level.

The highest level in the hierarchy is the image processing task to be performed. Examples are the recognition task or image enhancement. These tasks are performed by combining chips such as clocks and buffers with the basic image-processing chips from the second level.

The second level in the hierarchy contains the chips. Each chip is a complete functional block that performs a low-level vision task. Examples of chips are linear and logical convolvers, look-up tables, sorting functions, and contour tracers. Most of these chips accept a stream of input and perform the necessary delays required by two-dimensional image processing. Delays of 512 are required to align the rows of a 512 x 512 image when convolving spatially.

The third level is composed of macrocells. Macrocells are the blocks that compose the chips. Common functions include storage elements (RAMs, ROMs and line delays), bit-sliced data paths, and controllers. To speed the design, several tools automatically layout or assemble the bit-sliced data paths and program the ROM/PLAs.

The lowest level is composed of registers, adder cells, and ROM cells.

The images for the recognition system were obtained from a 512 by 512 "broadcast quality" video image with a frame rate of 30 Hz. Figure 11 shows an overview of the recognition system. Edge detection is the first stage in the image processing and, since it is the only processing step comparable to analog VLSI, the discussion will focus on it. The pattern matching and feature extracting stages are highly dependent on the assumption of two-dimensional images. This assumption is not consistent with a general vision system and as a consequence these two stages are not discussed here. The other processing step, contour tracing, although somewhat skew from the discussion of edge and discontinuity detection, is presented here briefly.

4.2 The Chips

The edge detection and contour tracing routines of Ruetz[30] are computed with several chips. The edge detection is performed by first smoothing the

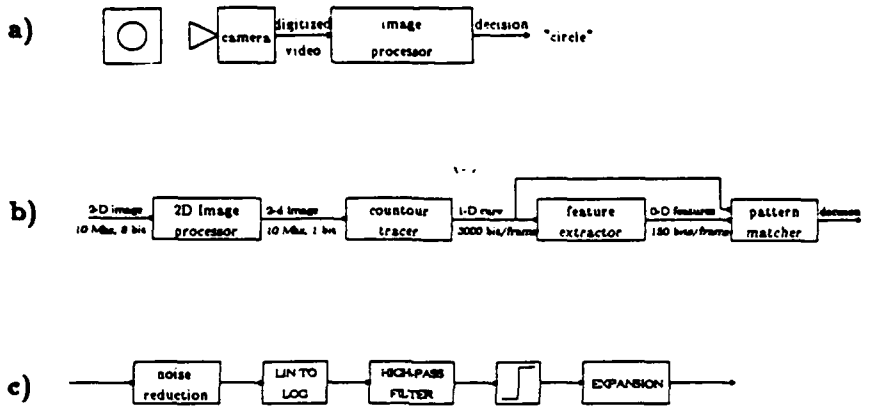

Figure 11: The image processing IC system[2]. a) An overview of the image processing system with camera and image processor. b) A block diagram overview of the image processor. c) The 2D image processor of Diagram b expanded into its component systems.

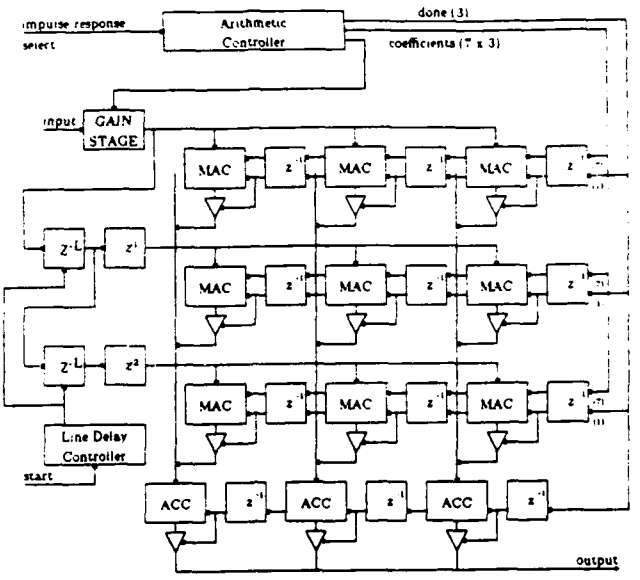

Figure 12: The 3 x 3 Linear Convolver[2].

input data with a low-pass filter to eliminate noise, high-pass filtering with a threshold to identify the edges, and then "bloating" and subsequently "eroding" the edges to produce closed contours. Each of these operations can be performed by convolving the input with a suitable mask. A 3x3 linear convolver chip was developed to perform the low and high pass filtering. The "bloating" and "eroding" were perform by a 7x7 logical convolver chip. In addition a contour tracing chip was developed. These chips are discussed in the following sections.

4.2.1 Convolution / Filtering

The convolver chip performs a real-time convolution on a 512 pixel / 512 line image with a 3 x 3 mask. The chip can be cascaded so that, for the binomial convolution discussed in Section 2.1.1, any binomial series can be produced. Besides binomial convolution, several other types of low-pass filters[30] can be utilized since the convolver chip has programmable coefficients.

Figure 12 shows a block diagram for the convolver chip. For all pixels $f_{i,j}$ in the image and for the FIR $h_{i,j}$, the convolver chip computes

$$g_{x,y} = \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} h_{i,j} f_{x-i,y-j}. \tag{13}$$

The convolution is performed not by shifting the image data; rather, the coefficients $h_{i,j}$ are shifted. Each of the accumulators (ACC's) (Figure 12, bottom) computes one complete convolution and, upon receipt of the "done" signal, outputs the result. The results appear at the clock rate with each accumulator's output delayed by one clock cycle (z^{-1}) from the previous accumulator. The ACC's perform three additions, one for each line in the convolution, to obtain the convolved result.

The arithmetic controller arranges for each row of three multiplying accumulators (MACs) to compute one line in the convolution. The line delay macrocells, z^{-L}, delay the image data by one line (512 pixels) for each separate row of MACs. The controller cycles through the coefficients for one row. For instance, the top row of MACs always computes the top row of the convolution since the coefficients appear as $\{h_{11}, h_{12}, h_{13}, h_{11}, \ldots\}$. Similarly the bottom row of MACs always computes the bottom row of the convolution, h_{3j} $j \in \{1,2,3\}$. The MACs within each row see the same data but have the coefficients $h_{i,j}$ delayed by one clock cycle. Like the ACCs, upon receipt of the "done" signal, a MAC outputs its result which is subsequently summed by an ACC. The MACs within a row produce an output at one third the pixel data rate.

Line Delay The function of the line delay is to accept one pixel and output the pixel delayed by one video line. Figure 13 shows the line delay architecture. The delay is implemented by shifting a pointer to the data rather than moving the data itself. Eight consecutive pixels of eight bits each are de-multiplexed to fill one 64 bit register. This register is then stored in a 63 x 64 bit RAM at, say, location n. The location n is incremented by 1 at one-eighth the pixel data rate (10 MHz). Simultaneous with writing the input register at site n, site $n+1$ in the RAM is read into the 64 bit output register. 8 bit chunks from the 64 bit output register are latched on to the output line. This implements the 512 pixel delay.

The line delay architecture has several advantages. The multiplexing

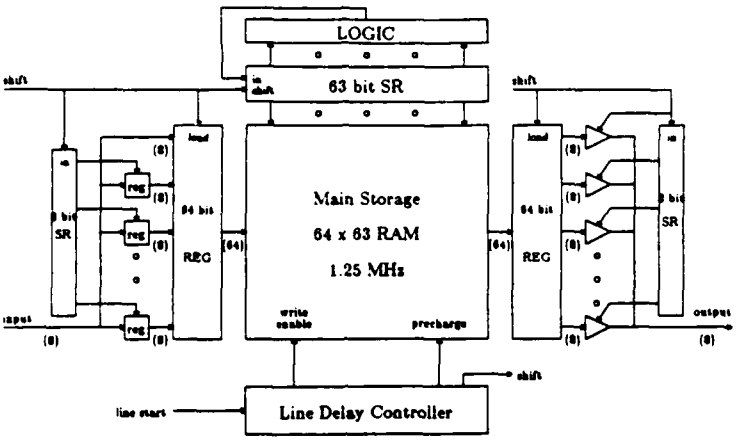

Figure 13: The line delay chip[2].

serves to reduce the 63 x 64 RAM rate to 1.25 MHz (for a 10MHz video signal). Consequently, lower power devices can be used. Also, the RAM is rectangular which makes laying out the line delay macrocells with the convolver MACs and ACCs simpler.

4.2.2 Dilation

Dilation and erosion are two computations performed on 1-bit edge maps. Dilation transforms a solitary one in the edge map into a region of ones. After the dilation, previously isolated ones may be connected to one another. This is one approach to filling gaps in edge detectors. After the thickening, the edge is then eroded until a thin contour is obtained. Ruetz[30] has developed a 7x7 logical convolver chip to perform this task.

The operation of dilation can be expressed as below.

$$g_{x,y} = OR_{n,m}(h_{n,m} AND f_{x-n,y-m}) \tag{14}$$

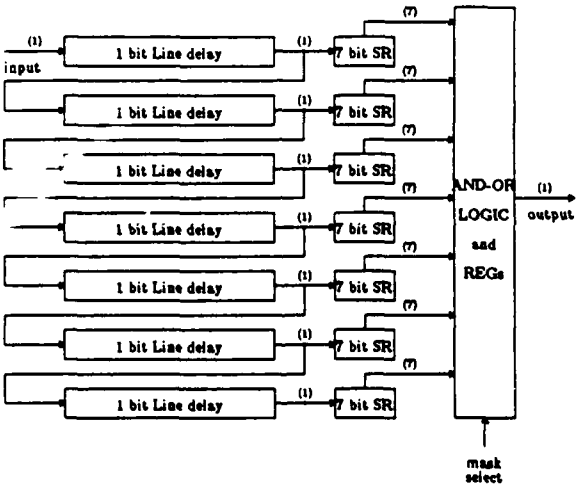

Figure 14: The 7 x 7 logical convolver chip[2].

$h_{n,m}$ is the 7x7, 1-bit dilation mask. This mask is simply ANDed with the delayed, 1-bit input data f. Figure 14 shows a block diagram of the logical convolver. The 7 1-bit delay lines are formed from the 8-bit delay line discussed previously by connecting the output on line n to the input on line $n+1$. The output from each of the 7 delay lines is stored in a 7 bit shift register. These 49 bits are then ANDed with the pre-stored mask and ORed together to obtain the final result.

4.2.3 Contour Tracing

Labeling contours is a common vision processing task. Once the contours in an image have been labeled, the contours can be broken into features such as straight lines, arcs, and corners and the length of the contour can be determined. Several recognition schemes require labeled edges[31, 32] and considerable effort has been spent on efficient contour following and labeling for the Connection Machine[33]. Most image contours are not simply biconnected and often will contain T-junctions and X-junctions[34].

Contour tracing is the first step for labeling contours. Ruetz simplifies the tracing problem by making several assumptions: each image contains one and only one contour, the contour is closed, and the contour is biconnected (no T or X junctions). With these assumptions a very simple, finite-state algorithm[35] for tracing can be employed[30].

Figure 15 show the architecture for the contour tracing chip. For real-time computation, the entire edge map must be buffered and, consequently, the 512x512 image was down-sampled to 128x120 pixels. The decimation function for the down-sampling could not be determined. To begin the tracing the controller searches for the first non-zero pixel by stepping through X and Y coordinates. Once this pixel is found, the controller checks pixels in its neighborhood in a deterministic order. The order ensures that, independent of contour direction, the contour will be traced. Once a neighboring pixel is found with a non-zero value, its δX and δY offsets are noted and the process repeats. With the starting (X,Y) position known, the series of offsets determines the path of the contour.

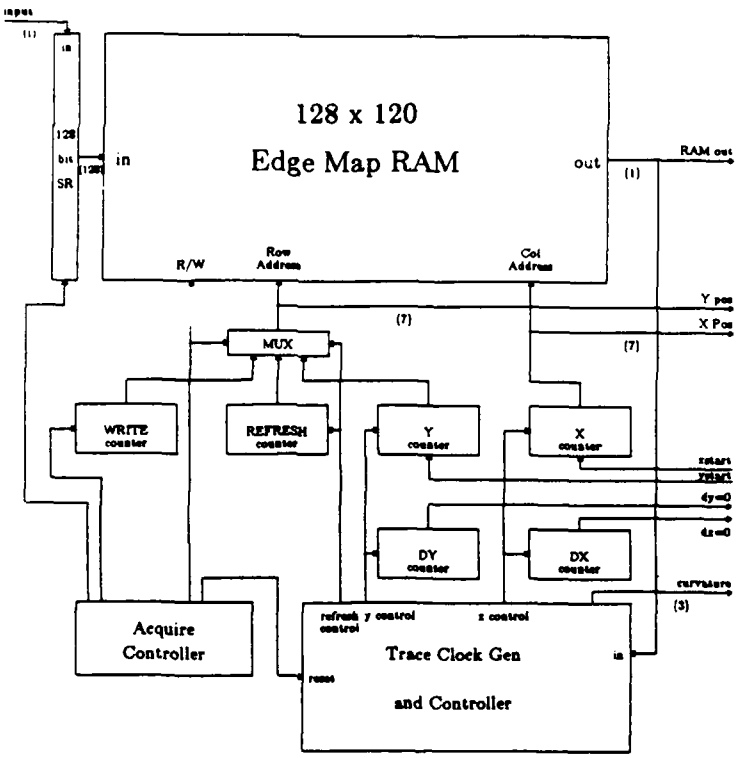

Figure 15: The contour tracer chip[2].

4.3 Advanced Chips for Image Processing

Since his work while at Berkeley, Ruetz has moved to LSI Logic Corp. which now produces several advanced image processing chips. The chips include 3 x 3 and 8 x 8 8-bit convolvers with a programmable FIR, line delay chips with variable delays of 512, 1024 or more, binary filters and template matchers, and rank-value filters. Some of these chips are briefly discussed below.

The Multi-bit Filter (L64240) from LSI Logic Corp. can perform two-dimensional convolution with an 8 x 8 window size at 20 MHz. For an 8 x 8 window, the input is 8 8-bit streams; the output is a 40-bit convolution over the window. The FIR coefficients are individually programmable. Inputs are provided to allow cascading of the chips, to facilitate increasing the window size, and to manipulate streams with more than 8-bits. In addition, the window shape can be re-configured to 1 x 64, 2 x 32, and 4 x 16, and the output can be scaled or delayed as desired. The chip price is roughly $1,300.

Another chip is the Variable-Length Video Shift Register (L64211). This chip takes an 8-bit input stream at 20 MHz and produces up to 8 8-bit outputs. Each output is delayed from the previous output by a length that is programmable between 12 and 516 pixels. This chip provides a means to shift the serial video signal by individual scan lines thereby providing the two-dimensional configuration for convolution. The cost is $115.

Figure 16 shows two possible configurations of 8 x 8 convolution chips and line delay chips to produce a 16 x 16 convolution. Another approach that avoids the expensive 8 x 8 convolvers is to cascade many 3 x 3 convolvers to build up the required window size. The cascading scheme uses less convolver chips for the same window size but is limited to masks that are the convolution of smaller masks and introduces a longer overall time delay. In addition, multiple scale output is available when chips are cascaded.

The Binary Filter and Template Matcher (L64230) is analogous to the 7 x 7 logical convolver discussed previously. This chip can perform the dilation and erosion computations in addition to pattern matching at 20 MHz. The chip can be configured as a 32 x 32 mask that requires 32 1-bit input lines. The output is 16 bits.

The fourth chip is the Rank-Value Filter (L64220). This filter sorts the pixels in an 8 x 8 (4 x 16, 2 x 32, or 1 x 64) size window and returns the pixel

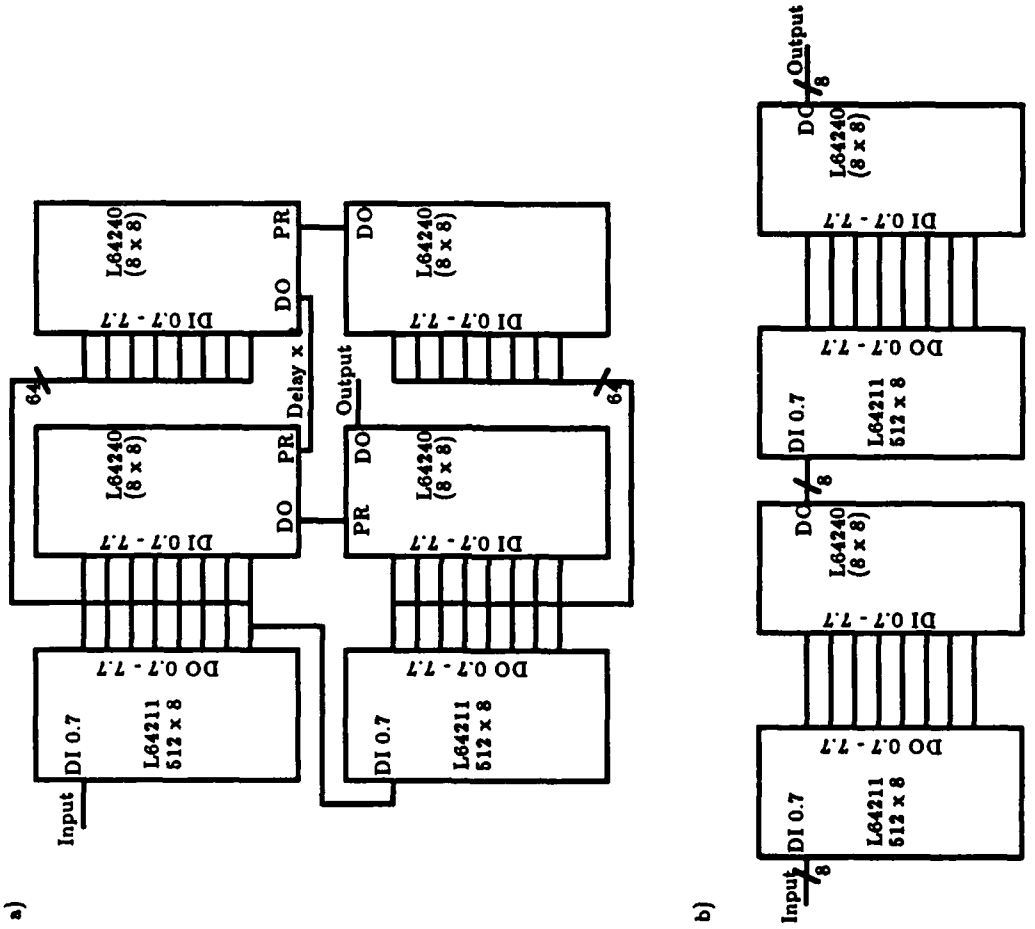

Figure 16: Two Multi-Bit Filter chip configurations for 16 x 16 convolution windows. a) A fully programmable 16 x 16 window with 40 bit output. b) Cascading two filters produces a 16 x 16 window from the convolution of the two 8 x 8 windows.

value of a specified rank. A typical use would be as a median, minimum, or maximum filter. The inputs and output are 12 bits and the chip can operate at up to 20 Mhz.

The L64720 (MCP) Video Motion Compensation processor computes the correlation between 16 x 16 (or 8 x 8) date blocks from two video images. For 16 x 16 data blocks, the L64720 achieves a 30 Hz performance on 352 x 288 images. The data blocks are correlated over an offset range of -8 to 7 pixels in both x and y. Additional devices can be used to increase the correlation offset range. This processor can implement the stereo[36] and motion[37] algorithms currently used on the Vision Machine[38]. The correlation is computed for every offset between the two data blocks and the offsets in x and y that minimizes the total of the absolute values of the differences between the data block pixels is returned. The cost for the L64720 is roughly $200.

LSI Logic sells two chips that are more general and useful then the contour following chip described by Ruetz[30]. One chip is a Histogram / Hough Transform Processor; the other chip implements contour tracing. The contour tracing chip can find all contours in an image. Output includes the slope (over 2 pixels) and curvature (over 3 pixels) of the contour as well as the "object" position, perimeter, and area. With the addition of external RAM, the contour tracer can process binary images measuring 1024 x 1024 pixels. Rectangular subregions of the image can be scanned to speed processing when, for instance, an object is being tracked. This contour following chip could serve as a preprocessor for line and curve finding algorithms and, ultimately, a recognition algorithm.

4.4 Discussion

For vision research, the primary inadequacy with the work of Ruetz is the limitation to two-dimensional objects; the vision problem solved is simplistic. As currently formulated, the general problem of vision requires analysis of the scene surface properties, such as depth and motion modules. These modules help the determination of object boundaries when images are complicated. The restriction by Ruetz to two dimensions makes his recognition problem solvable because the limited domain allows the use of simplifying assumptions.

The object edges found by Ruetz's edge detector are noiseless. This lack of noise is not a consequence of an efficient, optimized edge detector; rather the input image quality is so high that the edges produced are perfect. Ruetz shows examples of recognition results with images containing a single object. Each image has one and only one contour. Although T-junctions and X-junctions exist before the final processing stage, the final edge map does not provide any such junctions and the contour is closed. Not a realistic case, but it greatly simplifies the contour following and feature extraction process.

Ruetz's contour tracing chip immediately follows the edge detector. This is possible because the edges are noiseless. In a more sophisticated system, additional processing might be required, such as contour grouping and connecting, before the contour tracer would have a connected sequence of pixels.

Still, several aspects of Ruetz's work are significant. The 3x3 linear convolvers and line delay circuits are generally useful for early vision tasks. Many early vision problems are formulated to use local processing in order to increase the parallelism. While pixel-serial digital techniques are not highly parallel, the convolvers can perform the local processing demanded by the vision algorithms. For these digital chips, accuracy is not a problem. The chips use 8-bit data with accuracy no worse than most software implementations where 8 bits are used. Another significant aspect is the ability to cascade the convolvers thereby computing, for instance, binomial convolutions of any order. Unfortunately, for a 3 x 3 binomial convolution with coefficients $\{1/4, 1/2, 1/4\}$, in order to approximate a Gaussian with a standard deviation of 8, 127 3 x 3 convolutions must be performed. (Subsampling the image can significantly reduce the number of convolutions required to approximate a Gaussian with a large standard deviation.)

5 Analysis

Current analog vision processing has proceeded by mapping algorithms onto available VLSI circuit elements such as MOS transistors. Implementing a resistive network requires implementing a resistor. The resistor and its bias circuitry are made from MOS transistors as Figure 6 illustrates. Edge detection requires a resistive network and amplifier for a center-surround com-

putation. Discontinuity detection requires fuses. But ultimately everything is made from transistors. What is really needed is a novel hardware device that directly computes early vision tasks. This then would truly become "vision hardware" just as the the bipolar and horizontal cells are specialized for vision.

The question of local versus remote processing must be addressed at some point. Currently the analog VLSI developments at Caltech have utilized strictly local processing and, as was shown, these analog circuits scale poorly as the computational requirements increase. Eventually the optical quality will reach unacceptable levels with further increases in the computational requirements and, consequently the computational circuitry must be removed from the photoreceptor portion of the chip. Further, by removing the computational circuitry, special purpose, optical detectors, such as CCDs, could be employed.

Note that relocation of the processing to a remote location is precisely what evolution has provided for biological systems[39]. The early computational cells, horizontal cells, bipolar cells, etc, do not significantly reduce the resolution of the optical system. The resolution is primarily affected at the optic disk where the resolution is zero. The retina can maintain high resolution by utilizing three dimensions for the computational cells; integrated circuits have not been successful at utilizing three dimensions yet. The visual system for humans uses a large portion of the brain, yet the photoreceptors use only a tiny fraction. Given that the visual cortex is at the back of the brain, with an assumption that there is no remote processing and therefore computation is constrained to local processing only, our eyes would quite literally be located on the back of our heads. With this assumption, photoreceptors sparsely distributed throughout the visual cortex would clearly lead to unacceptable optical properties.

Remote processing was successfully implemented in the digital vision processing. The CCD imager was independent from the computational chips. Cascading some 3x3 convolvers to modify the image smoothing would not mandate modifying the CCD imager and its associated optical system. The technology that allows for the remote processing is the circuit speed. The circuits are fast enough to allow real-time processing even when serializing the image data. If more computation is required, the digital chip speed may not be fast enough to maintain real-time processing. The alternative is

to increase chip speed or begin to parallelize the digital computation. Note that the convolver chips already perform nine or more multiplications in parallel. Additional parallelism may be obtained from simultaneous computations on two or more pixels. Analog technology does not have a monopoly on parallel computation.

Other than fully serializing the data as digital techniques currently perform, remote processing is difficult to achieve. Consider two alternatives for a 512 x 512 imager: 1) process the image scan lines in parallel but serialize within a scan line[21], and 2) process all pixels in parallel. For the first alternative, 512 analog output lines are required. This number of output lines is well beyond present packaging technology. (Although as Yang[40] has demonstrated, 512 analog output lines may not be needed. The remote processing and area fan-out requirements can be achieved on the CCD chip. The binomial smoothing circuits are located adjacent to the CCD imager and occupy only at most 25% of the die size. This is an N-parallel, pipelined architecture as compared to the N x N parallel architecture of Mead and the serial, pipelined architecture of Ruetz.) The second alternative requires 512^2 analog output lines and is even more difficult to build. The technology that allows remote processing for these two cases is unrelated to speed. The need to access all the output lines is the dominant difficulty and consequently packaging technology plays a prominent role.

5.1 Parallel Hardware for Remote Processing

This final section presents some ideas related to packaging for parallel computation. The packaging must be designed to allow for remote processing of a large number of analog signals. Remote processing avoids the following problems inherent in strictly local processing: 1) reduction of optical resolution and efficiency, 2) non-modularity and non-expandability requiring complete chip redesign, and 3) fault-tolerant design for wafer-scale integration.

Ideally the imager should contain a large number and high density of photoreceptors. Assume that the imager is designed with criteria at the limit of processing technology. Define the area of a single photoreceptor to be a. At the limit of processing technology only a few devices can fit within the area a either on the imager or on another chip. Consequently, before any

processing can be performed there must be an *area fan-out.* For example, if area *a* can fit 4 MOS devices but an edge detector requires 20 devices, a fan-out of 5 is required.

One approach to remote-processing is to provide a pin on the chip carrier for every pixel or, if only lines are being parallelized, every line. For either situation there are too many pins with too high a density at too low a power. The size of the pixels requires that the wiring be VLSI.

The 3-D computer[41] suggests one possible scheme for remote processing. The 3-D computer stacks chips (each chip performs one processing task) using a 3-D wiring scheme. Each chip contains an array of N x N identical processing elements. Once stacked, each processing element in the array is connected to all the elements above and below it in the stack. This layout produces N x N stacks all working in parallel.

Microbridge interconnections[41] are used to stack the chips in the 3-D computer. The connections are made by tunneling through the chip substrate to the backside of the chip. The tunnel is highly doped to enable current transport between the chip circuits on the surface and the substrate's backside. A metal contact is made to the tunnel on the backside of one chip and the circuit-side of another chip. The two chips are then set on top of one another with the metal contacts touching. The microbridge interconnections are repeated throughout the array of processing elements and amongst all the chips.

The problem with this scheme is that there is no area fan-out. The processing is similar to that between the photoreceptor and the ganglion cells of the retina (at the fovea) rather than similar to the processing between the ganglion cells and the visual cortex. For certain calculations, like binomial convolution, no fan-out is required if each chip in the stack performs one convolution. Still, this 3-D computer does not meet the criteria for remote processing.

Another scheme for remote processing might be the *multichip module*[42] with a "flip-chip" imager. Figure 17 shows the multichip layout. The multichip itself is a wafer containing numerous flip-chip sites. Each flip-chip site contains an array of pads where metallic contact will be made between the flip-chip and the multichip module. Within the multichip are numerous levels of metallic leads separated by interlevel dielectrics. The dielectrics en-

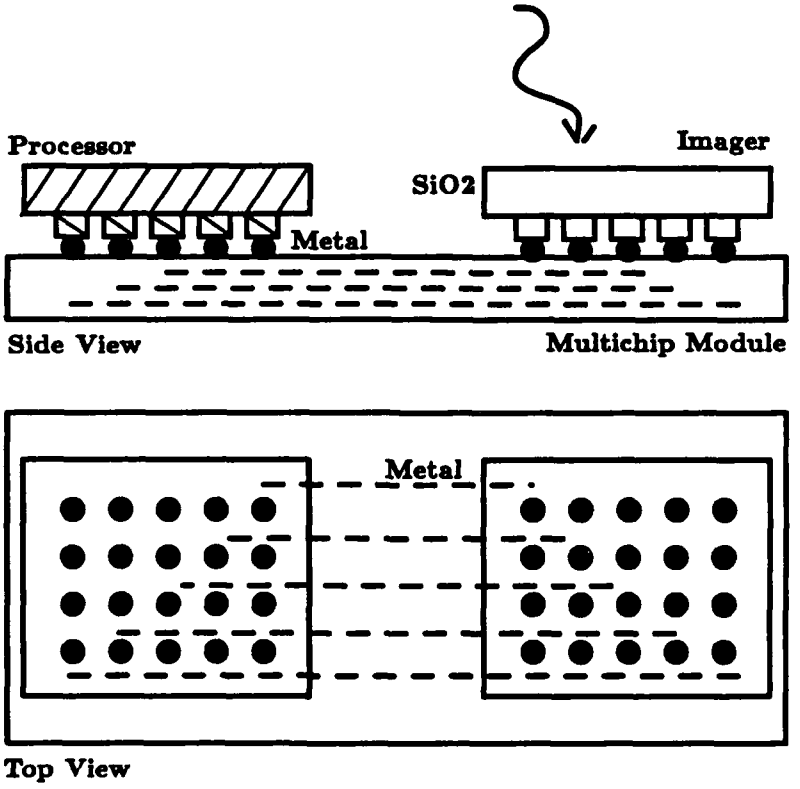

Figure 17: Multichip Modules with Flip Chip

sure that all metallic levels are electrically isolated. The metallic leads are arranged on the multichip to make connections between flip-chip pads. As many as 33 metallized layers have been fabricated with upwards of 12,000 chip pads[43]. This is the mechanism allowing remote processing with area fan-out.

Figure 17 also shows a proposal for an imager. The imager is a phototransistor array grown on a silicon dioxide substrate. The phototransistors are on the top of the substrate but, since silicon dioxide is transparent, light stimulates the phototransistor by passing through the substrate. Metallic contacts are placed on the top of the phototransistors; the chip is flipped and soldered to the pads on the multichip, maybe. Although not applied to an imager, flip-chips have been produced with 16,000 pads on a 128 x 128 array with solder bumps of 25 μm and inter-bump spacing of 60 μm[44].

The processing chips are arranged around the imager. The pads within any processing chip can have a density less than the pad density of the imager thereby allowing for area fan-out in the computation.

Note how this multichip scheme satisfies the requirements for remote processing. Modularity: the flip-chips can be individually designed, fabricated and tested. The pad pattern must be standardized. Fan-out: the available computational area grows with each multichip module. Wafer scale integration is not required for the complex processing and imaging chip. Only the relative simple multichip layout needs wafer scale techniques. Optical resolution is maintained. Some additional loses may exist due to substrate loses in the imager.

Ultimately techniques must be developed for remote processing if real-time vision systems are to be produced. Already the work of Carver Mead has highlighted some of the deficiencies ahead if local processing is adhered to. Developments in remote processing may require a long-term commitment to research but should pay off with biologically relevant, modular, and highly-parallel devices.

Digital techniques have achieved some success for real-time, 2D vision systems and, as an outgrowth, several useful chips, such a convolvers and line delays, are available. These chips can address some of the problems faced by early vision algorithms. Currently chip counts and cost may be high (for, say, binomial convolutions); however, the cost are dramatically less than

similar functionality (if available) in analog VLSI. Future development in digital image processing of convolvers with larger masks may help reduce chip costs and counts. Questions remain regarding the use of digital circuits for intermediate vision tasks. Still, digital circuits appear to be the best choice for implementation of vision algorithms in hardware today.

Continuing research in computational vision should remain committed to general-purpose, software systems. Hardware apparently will remain to inflexible, limiting, and costly for rapid development of computational theory. However, for some cases, such as parameter estimation with resistive fuses, the computational requirements are so immense that special purpose hardware, analog or digital, may illuminate the computational questions. Research in remote processing schemes should be undertaken concurrently with computational vision research to ensure a future, real-time hardware system for vision.

ACKNOWLEDGMENTS

This paper is based on an area exam during September 1989 in the Department of Electrical Engineering and Computer Science at M.I.T. Thanks to C. Sodini, H. Lee, and T. Poggio for serving on the examination committee and for providing helpful comments and suggestions. Also, thanks to B. Horn and T. Knight for helpful comments on drafts of this paper.

References

[1] Carver Mead. *Analog VLSI and Neural Systems.* Addison-Wesley, 1989.

[2] Peter A. Ruetz and Robert W. Broderson. Architectures and design techniques for real-time image-processing ic's. *IEEE Journal of Solid-State Circuits*, sc-22(2):233–250, April 1987.

[3] Berthold Horn, Hae-Sung Lee, James Little, Tomaso Poggio, Charles Sodini, and John Wyatt. Smart vision sensors: Analog VLSI systems for integrated image acquisition and early vision processing. *unpublished proposal*, March 1988.

[4] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* W.H. Freeman and Company, San Francisco, 1982.

[5] Edward B. Gamble and Tomaso Poggio. Visual integration and detection of discontinuities: The key role of intensity edges. A.I. Memo No. 970, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, October 1987.

[6] Jan Van der Spiegel, G. Kreider, C. Claeys, I. Debusschere, G. Sandini, P. Dario, F. Fantini, P. Bellutti, and G. Soncini. A foveated retina-like sensor using CCD technology. In Carver Mead and Mohammed Ismail, editors, *Analog VLSI Implementation of Neural Systems*, pages 189–212. Kluwer Academic Publishers, London, 1989.

[7] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London*, B(207):187–217, 1980.

[8] Berthold K.P. Horn. Parallel networks for machine vision. A.I. Memo No. 1071, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, December 1988.

[9] Tomaso Poggio, Vincent Torre, and Christof Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.

[10] Tomaso Poggio and C. Koch. Ill-posed problems in early vision: from computational theory to analog networks. *Proceedings of the Royal Society of London B*, 226:303–323, 1985.

[11] Demetri Terzopoulos. *Multiresolution Computation of Visible-Surface Representations.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1984.

[12] Berthold K. P. Horn. *Robot Vision.* MIT Press, Cambridge, Mass., 1986.

[13] Thomas F. Knight Jr. *Design of an Integrated Optical Sensor with On-Chip Preprocessing.* PhD thesis, Massachusetts Institute of Technology, 1983.

[14] Stuart Geman and Don Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6:721–741, 1984.

[15] Jose L. Marroquin. *Probabilistic Solution of Inverse Problems.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1985.

[16] Jose L. Marroquin, Sanjoy Mitter, and Tomaso Poggio. Probabilistic solution of ill-posed problems in computational vision. *J. Amer. Stat. Assoc.*, 82:76–89, 1987.

[17] Andrew Blake and Andrew Zisserman. *Visual Reconstruction.* MIT Press, Cambridge, Mass, 1987.

[18] J. Hutchinson, C. Koch, J. Luo, and C. Mead. Computing motion using analog and binary resistive networks. *IEEE Computer Magazine*, 21:52–64, 1988.

[19] J.P. Sage and A. Lattes. A high-speed analog two-dimensional gaussian image convolver. In *Technical Digest of the Optical Society of America Topical Meeting on Machine Vision*, Incline Village, NV, March 1985.

[20] Woodward Yang. A charge-coupled device architecture for on focal plane image signal processing. In *International Symposium on VLSI Technology, Systems, and Applications*, Taipei, Taiwan, May 1989.

[21] Woodward Yang and Alice M. Chiang. Vlsi processor architectures for computer vision. In *Proceedings Image Understanding Workshop*, Palo Alto, CA, May 1989. Morgan Kaufmann, San Mateo, CA.

[22] Thomas F. Knight Jr. private communication, 1989.

[23] M.A. Mahowald and C.A. Mead. Silicon retina. In Carver Mead, editor, *Analog VLSI and Neural Systems*, pages 257–278. Addison-Wesley, 1989.

[24] John Harris, Christof Koch, Jin Luo, and John Wyatt. Resistive fuses: Analog hardware for detecting discontinuities in early vision. In Carver Mead and Mohammed Ismail, editors, *Analog VLSI Implementation of Neural Systems*, pages 27–56. Kluwer Academic Publishers, London, 1989.

[25] Carver Mead. Adaptive retina. In Carver Mead and Mohammed Ismail, editors, *Analog VLSI Implementation of Neural Systems*, pages 239–246. Kluwer Academic Publishers, London, 1989.

[26] Christof Koch. Seeing chips: Analog VLSI circuits for computer vision. *Neural Computation*, 1:184–200, 1989.

[27] Lance A. Glasser. A uv write-enabled prom. In *1985 Chapel Hill Conference on VLSI*, pages 61–65, Rockville, MD, 1985. Computer Science Press.

[28] E. B. Gamble Jr. *Integration of Vision Modules for Recognition.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1990.

[29] M.A. Mahowald and T. Delbruck. Cooperative stereo matching using static and dynamic image features. In Carver Mead and Mohammed Ismail, editors, *Analog VLSI Implementation of Neural Systems*, pages 213–238. Kluwer Academic Publishers, London, 1989.

[30] Peter A. Ruetz. *Architectures and Design Techniques for Real-Time Image-Processing IC's.* PhD thesis, University of California, Berkeley, CA, 1986.

[31] Todd A. Cass. Parallel computation in model-based recognition. Master's thesis, Massachusetts Institute of Technology, June 1988.

[32] Daniel P. Huttenlocher. *Three-Dimensional Recognition of Solid Objects from a Two-Dimensional Image.* PhD thesis, Massachusetts Institute of Technology, 1989.

[33] James J. Little, Guy E. Blelloch, and Todd Cass. Algorithmic techniques for vision on a fine-grained parallel machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence, in press*, 1989.

[34] David Beymer. Junctions: their detection and use for grouping in images. Master's thesis, Massachusetts Institute of Technology, 1989.

[35] P.W. Kitchin et. al. Processing of binary images. In *Robot Vision.* IFS Ltd., UK, 1983.

[36] Walter Gillett. Issues in parallel stereo matching. Master's thesis, Massachusetts Institute of Technology, 1988.

[37] Heinrich H. Bülthoff, James J. Little, and Tomaso Poggio. A parallel algorithm for real-time optical flow. *Nature*, 337:549 – 553, February 1989.

[38] Tomaso Poggio, J. Little, E. Gamble, W. Gillett, D. Geiger, D. Weinshall, M. Villalba, N. Larson, T. Cass, H. Bülthoff, M. Drumheller, P. Oppenheimer, W. Yang, and A. Hurlbert. The MIT Vision Machine. In *Proceedings Image Understanding Workshop*, Cambridge, MA, April 1988. Morgan Kaufmann, San Mateo, CA.

[39] Stephen W. Kuffler, John G. Nicholls, and A. Robert Martin. *From Neuron to Brain: A Cellular Approach to the Function of the Nervous System.* Sinauer Associates Inc, Sunderland, MA, 1984.

[40] Woodward Yang. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1990 (in preperation).

[41] Jan Grinberg, Graham R. Nudd, and R. David Etchells. A cellular VLSI architecture. *IEEE Computer*, pages 69–81, January 1984.

[42] Rao R. Tummala and Eugene J. Rymaszewski. *Microelectronics Packaging Handbook.* Van Nostrand Reinhold, New York, 1989.

[43] A. J. Blodgett and D. R. Barbour. Thermal conduction module: A high-performance multilayer ceramic package. *IBM Journal of Research and Development*, 26(1):30–36, January 1982.

[44] W. Weston. High density 128 x 128 area arrays of vertical electrical interconnections. In *4th Annual Microelectronic Interconnect Conference*, July 1985.